WHAT IS CLAIMED IS:

1.   Apparatus for dynamically transforming and caching at least one computer program, the apparatus comprising:

      a.   one or more computer readable storage media; and

      b.   computer executable instructions stored in the one or more computer readable storage media, the computer executable instructions comprising:

            i.   instructions for dynamically transforming code fragments;

            ii.   instructions for caching said code fragments;

            iii. instructions for causing said code fragments to be executed by at least one computer processor; and

            iv.   instructions providing an application programming interface enabling said at least one computer program to activate said instructions for dynamically transforming said code fragments and said instructions for caching said code fragments.

2.   The apparatus of claim 1, wherein said instructions providing an application programming interface enable said at least one computer program to provide said code fragments for said instructions for dynamically transforming code fragments and for said instructions for caching said code fragments.

3.   The apparatus of claim 1, wherein said instructions providing an application programming interface include providing functions for caching and executing a specified code fragment.

4.   The apparatus of claim 1, wherein said instructions providing an application programming interface include providing functions for configuring behavior of said instructions for dynamically transforming said code fragments and said instructions for caching said code fragments.

5.   The apparatus of claim 1, wherein said instructions for dynamically transforming said code fragments comprise instructions for changing memory address references in said code fragments.

6.   The apparatus of claim 1, wherein said instructions for dynamically transforming said code fragments comprise instructions for changing the layout of said code fragments while preserving the function of said code fragments.

7.   The apparatus of claim 1, wherein said instructions providing an application programming interface include instructions for accessing code fragments across a network.

8.   The apparatus of claim 1, wherein said at least one computer program comprises at least one emulator.

9.   The apparatus of claim 1, wherein said at least one computer program comprises a plurality of emulators.

10.   The apparatus of claim 9, wherein said plurality of emulators comprise emulators for at least two different computer architectures.

11. The apparatus of claim 1, wherein said at least one computer program comprises at least one operating system.

12. The apparatus of claim 1, said computer executable instructions further comprising instructions for transparently obtaining said code fragments from said at least one computer program for said instructions for dynamically transforming said code fragments and for said instructions for caching said code fragments.

13. The apparatus of claim 12, said computer executable instructions further comprising instructions for controlling the execution of said at least one computer program on said at least one computer processor.

14. The apparatus of claim 12, said computer executable instructions further comprising instructions for obtaining optimal portions of code from said at least one computer program to create said code fragments.

15. The apparatus of claim 12, wherein said instructions for transparently obtaining said code fragments from said at least one computer program obtain said code fragments across a network.

16. The apparatus of claim 1, said computer executable instructions further comprising instructions for optimizing said code fragments.

17. The apparatus of claim 1, said computer executable instructions further comprising instructions for linking a plurality of said code fragments to create a larger code fragment.

18. The apparatus of claim 1, said computer executable instructions further comprising instructions for replacing hardware control code in said code fragments, where said hardware control code is adapted to control

5 hardware which not present and hardware which is not functioning.

19. Apparatus for dynamically transforming and caching at least one computer program, the apparatus comprising:

    a.   one or more computer readable storage media; and

5   b.   computer executable instructions stored in the one or more computer readable storage media, the computer executable instructions comprising:

    i.   instructions for dynamically transforming code fragments;

10   ii. instructions for dynamically optimizing said code fragments;

    iii. instructions for caching said code fragments;

    iv. instructions for changing hardware

15 control code in said code fragments;

    v.   instructions for transparently obtaining said code fragments from said at least one computer program;

    vi. instructions providing an application

20 programming interface enabling said at least one computer program to activate said instructions for dynamically transforming and caching said code fragments; and

    vii. instructions for causing said code

25 fragments to be executed by at least one computer processor.

-31-

20. Apparatus for executing a plurality of software
applications, the apparatus comprising:

30   a.   one or more computer readable storage
media; and

            b.   computer executable instructions stored in
the one or more computer readable storage media, the
computer executable instructions comprising:

35   i.   instructions for obtaining portions
of computer program code from said plurality of
software applications;

                ii.   instructions for dynamically
transforming and caching said portions of
40   computer program code to create transformed
code fragments; and

                iii. instructions for executing said
transformed code fragments.

21. The apparatus of claim 20, wherein said plurality of
software applications comprise emulators.

22. The apparatus of claim 21, wherein said plurality of
software applications emulate computer systems with at
least two different instruction set architectures.